

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-27102

(43) 公開日 平成10年(1998) 1月27日

(51) Int.Cl.⁶

G 0 6 F 9/38

識別記号

3 5 0

庁内整理番号

F I

G 0 6 F 9/38

技術表示箇所

3 5 0 X

審査請求 未請求 請求項の数11 O L (全 10 頁)

(21) 出願番号 特願平8-182141

(22) 出願日 平成8年(1996) 7月11日

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 寺田 光一

神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内

(72) 発明者 小島 啓二

神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内

(72) 発明者 藤川 義文

神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内

(74) 代理人 弁理士 富田 和子

最終頁に続く

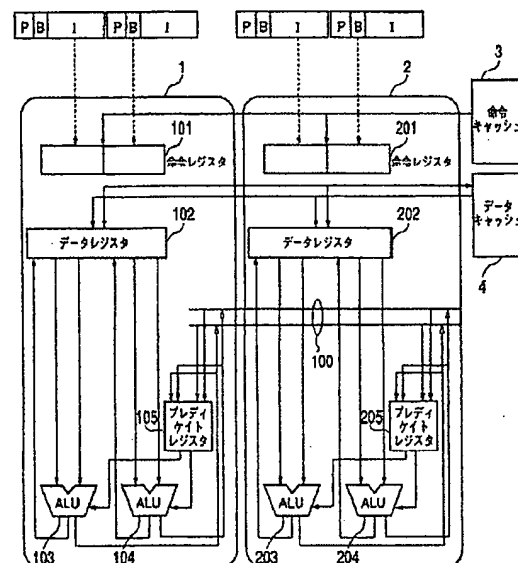
(54) 【発明の名称】 演算処理装置

(57) 【要約】

【課題】 V L I W型の演算処理装置の特徴と、パイプライン型の演算処理装置の特徴とを兼ね備えた演算処理装置を構成する場合、並列に演算処理を実行している複数の演算器の相互間で演算結果の参照を行うために、演算器相互間でのレジスタファイルの転送処理が頻繁に発生し、高速化の効果を十分に得られないことがある。これを解決する。

【解決手段】 プレディケイトレジスタ105、205を設け、オペランド指定により、プレディケイトレジスタの更新データを全演算器1、2に対して報知するための手段を設ける。これにより、条件分岐演算と数値演算とをそれぞれ異なる演算器で演算することがオーバーヘッド無しで可能になるため処理プログラムのステップ数を小さくできる。また、ビット幅が大きいデータレジスタの高速な演算器間転送が不要になり、実装面積を小さくできるため高速な演算処理装置を実現できる。

図 1



【特許請求の範囲】

【請求項 1】 1 命令語中に複数の演算命令フィールドを持ち、複数の演算の処理を並列に実行する演算処理装置において、

前記複数の演算命令フィールドを、それぞれが少なくとも 1 つの演算命令フィールドを含む複数の演算命令フィールドグループに分け、

これら複数の演算命令フィールドグループにそれぞれ対応して設けられ、それぞれが対応する演算命令フィールドグループでのみ評価される複数のレジスタと、

前記複数のレジスタに格納されている値の評価に基づいて演算を実行するかどうかを選択する演算回路と、

前記複数のレジスタに演算結果に基づいて値を格納する手段とを具備し、

前記複数のレジスタに値を格納する手段は、

それぞれが対応する演算命令フィールドグループでの演算結果を評価した値と、それぞれが対応する演算命令フィールドグループ以外での演算結果を評価した値とを、選択的に格納する手段であることを特徴とする演算処理装置。

【請求項 2】 1 命令語レジスタ中に複数の演算命令フィールドを持ち、前記複数の演算の処理を並列に実行する演算処理装置において、

前記複数の演算命令フィールドを、それぞれが少なくとも 1 つの演算命令フィールドを含む複数の演算命令フィールドグループに分け、

これら複数の演算命令フィールドグループにそれぞれ対応して設けられ、いずれかの演算命令フィールドグループで評価される複数のレジスタと、

前記複数のレジスタに演算結果に基づいて値を格納する手段と、

前記複数のレジスタに格納されている値を評価する手段と、

前記複数のレジスタに格納されている値の評価に基づいて演算を実行するかどうかを選択する演算回路とを具備し、

前記複数のレジスタに格納されている値を評価する手段は、

それぞれが対応する演算命令フィールドグループの前記レジスタに格納されている値と、

それぞれが対応する演算命令フィールドグループ以外の前記レジスタに格納されている値とを、選択的に評価する手段であることを特徴とする演算処理装置。

【請求項 3】 請求項 1 に記載の演算処理装置であって、前記複数のレジスタに値を格納する手段は、

前記複数の演算命令フィールドグループの全てに向けて演算結果を評価した値が報知されている場合には、該報知されている値を選択して格納し、

そうでない場合には、それぞれが対応する演算命令フィールドグループでの演算結果を評価した値を選択して格

納する手段であることを特徴とする演算処理装置。

【請求項 4】 請求項 2 に記載の演算処理装置であって、前記複数のレジスタに格納されている値を評価する手段は、

前記複数の演算命令フィールドグループの全てに向けて前記レジスタのいずれかに格納されている値が報知されている場合には、該報知されている値を選択して評価し、

そうでない場合には、それぞれが対応する演算命令フィールドグループでの前記レジスタに格納されている値を選択して評価する手段であることを特徴とする演算処理装置。

【請求項 5】 請求項 3 に記載の演算処理装置であって、前記演算命令フィールドは、

前記複数の演算命令フィールドグループの全てに向けて演算結果を評価した値を報知するかどうかを決定するための演算命令ビット列を内包していることを特徴とする演算処理装置。

【請求項 6】 請求項 4 に記載の演算処理装置であって、前記演算命令フィールドは、

前記複数の演算命令フィールドグループの全てに向けて前記複数のレジスタいずれかに格納されている値を報知するかどうかを決定するための演算命令ビット列を内包していることを特徴とする演算処理装置。

【請求項 7】 請求項 3 または 5 に記載の演算処理装置であって、

前記複数の演算命令フィールドグループのうち 2 つ以上の演算命令フィールドグループから、前記複数の演算命令フィールドグループの全てに向けて演算結果を評価した値が報知されていることを検出する手段と、

前記検出の結果に基づいて例外信号を発生させる手段とを具備することを特徴とする演算処理装置。

【請求項 8】 請求項 4 または 6 に記載の演算処理装置であって、

前記複数の演算命令フィールドグループのうち 2 つ以上の演算命令フィールドグループから、前記複数の演算命令フィールドグループの全てに向けていずれかの前記レジスタに格納されている値が報知されていることを検出する手段と、

前記検出の結果に基づいて例外信号を発生させる手段とを具備することを特徴とする演算処理装置。

【請求項 9】 請求項 3、4、5、6、7 または 8 に記載の演算処理装置であって、

前記値を報知する手段は、共有バス線による報知を行う手段であることを特徴とする演算処理装置。

【請求項 10】 1 命令語中に複数の演算命令フィールドを持ち、複数の演算の処理を並列に実行する演算処理装置において、

それぞれ、前記複数の演算命令フィールドの少なくとも 1 つの演算命令フィールドに対応して設けられた複数の

演算器を備え、
各演算器は、
前記少なくとも 1 つの演算命令フィールドを保持する命令レジスタと、
前記少なくとも 1 つの演算命令フィールドに対応して演算を行う演算回路と、
該演算回路の演算を実行するか否かを決定する値を格納するレジスタとを有し、
予め定められた命令について、その命令の演算結果を評価した値をすべての前記演算器内のレジスタに書き込む手段をさらに備え、
前記各演算器内の演算回路は、前記レジスタを指定された演算命令について、当該レジスタに書き込まれた値に応じて当該演算命令を実行するか否かを決定することを特徴とする演算処理装置。

【請求項 11】 1 命令語中に複数の演算命令フィールドを持ち、複数の演算の処理を並列に実行する演算処理装置において、
それぞれ、前記複数の演算命令フィールドの少なくとも 1 つの演算命令フィールドに対応して設けられた複数の演算器を備え、
各演算器は、
前記少なくとも 1 つの演算命令フィールドを保持する命令レジスタと、
前記少なくとも 1 つの演算命令フィールドに対応して演算を行う演算回路と、
該演算回路の演算を実行するか否かを決定する値を格納するレジスタと、
予め定められた命令について、その命令の演算結果を評価した値を自己の演算器内のレジスタに書き込む手段とを有し、
前記各演算器内の演算回路は、いずれかの演算器内の前記レジスタを指定された演算命令について、当該レジスタに書き込まれた値に応じて当該演算命令を実行するか否かを決定することを特徴とする演算処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は 1 命令語レジスタ中に複数の演算命令フィールドを持ち、これらの演算命令を並列に実行する演算処理装置に関する。

【0002】

【従来の技術】 伝統的な演算処理装置は、1 命令語で 1 つの演算処理を行うように構成され、命令語の列は直列的に逐一演算されるのが一般的であった。

【0003】 これに対し近年になって、実行速度の向上のために 1 つの命令語で複数の演算命令を処理するようにした命令体系を持ち、これら複数の演算命令を並列に実行する演算処理装置が開発されてきた。これらの演算処理装置は総称して、VLIW (Very Long Instruction Word) 型演算処理装置と

呼ばれている。

【0004】 この型の演算処理装置は、並列に複数の演算命令を実行するために、複数の演算器を持つ。また、複数の演算器に対応した複数のレジスタファイルを持ち、それぞれの演算器が独立して演算処理を実行できるように構成することが多い。このような複数の演算器を用いて特定の演算処理を行う場合、一般に、演算器相互間のデータ通信が不可欠である。この目的のため、この型の演算処理装置は、例えば、複数の演算器間でのレジスタ値転送手段、もしくは、複数の演算器からアクセス可能な共用レジスタのような手段を持つ。このような演算処理装置としては、例えば、特開平 5-233281 号公報に開示されているような技術がある。

【0005】 また、以上述べたような高速化手段とは別に、演算処理自体を時系列に複数のステージに分割し、複数の独立したステージがそれぞれ直列的に演算処理を実行する演算処理装置がある。これらの演算処理装置は、パイプライン型演算処理装置と呼ばれている。

【0006】 この型の演算処理装置は、命令語が直列的に並んでいる場合に、最大の性能を発揮できることが知られている。これに対して命令語が直列的に並ばずに、例えば、条件分岐命令が含まれる演算処理の場合は、パイプラインの制御が乱れ、一時的な性能の低下が発生する。

【0007】 このような問題を克服するために、この型の演算処理装置は、条件分岐処理を減らすための改良が行われている。その方法の 1 つは、プレディケイトレジスタ (Predicate Register) を用いる方法である。

【0008】 プレディケイトレジスタは、命令語を修飾し、当該命令語を実行処理するかどうかを決定するためのレジスタである。プレディケイトレジスタを用いることで、条件分岐命令の使用頻度を大幅に減らすことが可能になる。後述する発明の理解のため、この点についての簡単な説明を図面を用いて行う。

【0009】 図 2 は C 言語によるプログラム例を、図 3 は図 2 のプログラムを従来型の演算処理装置に与える形式にコンパイルした例を、図 4 は図 2 のプログラムをプレディケイトレジスタ型の演算処理装置に与える形式にコンパイルした例をそれぞれ示している。図に示すように、図 3 では条件分岐によって実現される演算処理が、図 4 では条件分岐を必要とせずに実現できる。図 4 の 2 行目が、プレディケイトレジスタを用いた命令語の記述を表している。1 行目の比較命令によって、1 番目のプレディケイトレジスタ (p0) に比較結果の値を書き込む。2 行目の減算命令は、命令語に前置されている記述「(p0)」によって、p0 に格納されている値が「真」である場合にだけ減算処理が実行される。p0 に格納されている値が「偽」である場合は、2 行目の減算命令は空読みされて実行処理は行われない。このような

実行方法により、条件分岐を減らすことができる。

【0010】

【発明が解決しようとする課題】しかし、上記のような演算処理の高速化手段を併用して用いる場合、即ち、VLIW型の演算処理装置の特徴と、パイプライン型の演算処理装置の特徴とを兼ね備えた演算処理装置を構成する場合、次のような課題がある。

【0011】並列に演算処理を実行している複数の演算器の相互間で、演算結果の参照を行うために、演算器相互間でのレジスタファイルの転送処理が頻繁に発生し、結果的に並列処理による高速化やパイプライン処理による高速化の効果を十分に得られないことがあった。

【0012】本発明の目的は、1命令語中に複数の演算命令フィールドを持ち、複数の演算の処理を並列に実行する演算処理装置において、複数の演算を実行するための複数の演算器の相互間での、演算結果の転送処理の回数を削減することで、処理プログラムのステップ数を小さくすることができる演算処理装置を提供することにある。

【0013】

【課題を解決するための手段】本発明による演算処理装置は、1命令語中に複数の演算命令フィールドを持ち、複数の演算の処理を並列に実行する演算処理装置において、前記複数の演算命令フィールドを、それぞれが少なくとも1つの演算命令フィールドを含む複数の演算命令フィールドグループに分け、これら複数の演算命令フィールドグループにそれぞれ対応して設けられ、それぞれが対応する演算命令フィールドグループでのみ評価される複数のレジスタと、前記複数のレジスタに格納されている値の評価に基づいて演算を実行するかどうかを選択する演算回路と、前記複数のレジスタに演算結果に基づいて値を格納する手段とを具備し、前記複数のレジスタに値を格納する手段は、それぞれが対応する演算命令フィールドグループでの演算結果を評価した値と、それぞれが対応する演算命令フィールドグループ以外での演算結果を評価した値とを、選択的に格納する手段であることを特徴とする。

【0014】また、本発明による他の演算処理装置は、1命令語レジスタ中に複数の演算命令フィールドを持ち、前記複数の演算の処理を並列に実行する演算処理装置において、前記複数の演算命令フィールドを、それぞれが少なくとも1つの演算命令フィールドを含む複数の演算命令フィールドグループに分け、これら複数の演算命令フィールドグループにそれぞれ対応して設けられ、いずれかの演算命令フィールドグループで評価される複数のレジスタと、前記複数のレジスタに演算結果に基づいて値を格納する手段と、前記複数のレジスタに格納されている値を評価する手段と、前記複数のレジスタに格納されている値の評価に基づいて演算を実行するかどうかを選択する演算回路とを具備し、前記複数のレジスタ

に格納されている値を評価する手段は、それぞれが対応する演算命令フィールドグループの前記レジスタに格納されている値と、それぞれが対応する演算命令フィールドグループ以外の前記レジスタに格納されている値とを、選択的に評価する手段であることを特徴とする。

【0015】他の観点によれば、本発明による演算処理装置は、1命令語中に複数の演算命令フィールドを持ち、複数の演算の処理を並列に実行する演算処理装置において、それぞれ、前記複数の演算命令フィールドの少なくとも1つの演算命令フィールドに対応して設けられた複数の演算器を備え、各演算器は、前記少なくとも1つの演算命令フィールドを保持する命令レジスタと、前記少なくとも1つの演算命令フィールドに対応して演算を行う演算回路と、該演算回路の演算を実行するか否かを決定する値を格納するレジスタとを有し、予め定められた命令について、その命令の演算結果を評価した値をすべての前記演算器内のレジスタに書き込む手段をさらに備え、前記各演算器内の演算回路は、前記レジスタを指定された演算命令について、当該レジスタに書き込まれた値に応じて当該演算命令を実行するか否かを決定することを特徴とする。

【0016】さらに他の観点によれば、本発明による演算処理装置は、1命令語中に複数の演算命令フィールドを持ち、複数の演算の処理を並列に実行する演算処理装置において、それぞれ、前記複数の演算命令フィールドの少なくとも1つの演算命令フィールドに対応して設けられた複数の演算器を備え、各演算器は、前記少なくとも1つの演算命令フィールドを保持する命令レジスタと、前記少なくとも1つの演算命令フィールドに対応して演算を行う演算回路と、該演算回路の演算を実行するか否かを決定する値を格納するレジスタと、予め定められた命令について、その命令の演算結果を評価した値を自己の演算器内のレジスタに書き込む手段とを有し、前記各演算器内の演算回路は、いずれかの演算器内の前記レジスタを指定された演算命令について、当該レジスタに書き込まれた値に応じて当該演算命令を実行するか否かを決定することを特徴とする。

【0017】上記の各構成によれば、複数の演算を実行するための複数の演算器の相互間での、演算結果の転送処理の回数を削減することができる。その結果、処理プログラムのステップ数を小さくすることが可能になる。

【0018】

【発明の実施の形態】以下、図面を参照して、本発明の実施の形態について説明する。

【0019】図5は、本発明の一実施の形態の演算処理装置の概略構成を示している。図5において、1及び2は演算器、3は命令キャッシュ、4はデータキャッシュである。図に示す他に、主記憶部、入出力部などの構成要素を持つが、これらは本発明に関係なく、また周知の技術で構成可能であるため、省略している。

【0020】本実施の形態での命令語は、図6に示すように、演算命令1フィールドと、演算命令2フィールドとから構成される。演算命令1フィールドと演算命令2フィールドはそれぞれ、プレディケイトレジスタ選択フィールド(Pフィールド)と、プレディケイトレジスタ報知フィールド(Bフィールド)と、命令フィールド(Iフィールド)とから構成される。図6において、601は命令語、611は演算命令1フィールド、612は演算命令2フィールド、621及び631はPフィールド、622及び632はBフィールド、623及び633はIフィールド、をそれぞれ示す。これらのフィールドのうち、演算命令1フィールドは演算器1に供給され、演算命令2フィールドは演算器2に供給される。

【0021】また、ここで示した命令語の構成は、あくまでも例であり、同じ要旨であれば構成を変えて実施しても構わない。例えば、演算命令フィールドは2つである必要はなく、より多く、例えば4つであっても良い。

【0022】図5に戻り、演算器1は次の構成要素からなる。101は命令レジスタであり、命令キャッシュ3から読み出された命令のうち演算器1に関する演算命令1フィールドを保持する。102はデータレジスタであり、演算器1において演算処理を行うためのオペランドデータを保持する。103は、演算命令1フィールドによって指定された命令を実行するためのALUである。105は、ALU103が命令を実行するかどうかの情報を保持するプレディケイトレジスタである。106は、プレディケイトレジスタ105の値を更新する際に入力データを選択するためのセレクトである。

【0023】演算器2は、演算器1と同じ構成要素からなる。即ち演算器2は、命令レジスタ201、データレジスタ202、ALU203、プレディケイトレジスタ205、セレクト206とから構成される。

【0024】ここで、データレジスタ102及び202、プレディケイトレジスタ105及び205は、それぞれが複数のレジスタ領域を持ち、それらを選択して利用できる。例えば、32個のデータレジスタ領域を持つ場合、その中から一つのデータレジスタ領域を選択するために、5本のデータレジスタ領域選択信号を、データレジスタ102及び202にそれぞれ入力することになる。この場合、各データレジスタ領域を識別するために、例えば、r0、r1、r2のように、データレジスタ領域に番号を割り振ることが一般に行われている。なお以上説明した内容は、データレジスタだけでなく、プレディケイトレジスタに対しても全く同じように適用される。

【0025】次に本実施の形態の動作について説明する。なお、ここで述べる以外の通常のALUを用いた演算については、従来より周知の技術で実現できるため、ここでは説明を省略する。

【0026】初めに、図4に示す命令列を実行する際の

動作について説明する。図4は3語からなる命令列の例であり、演算器を一つだけ用いる場合の例である。

【0027】第一の命令語は比較命令(cmp.gt)であって、データレジスタ領域1(r1)と即値"2"とを比較し、データレジスタ領域1の方が大きい場合にプレディケイトレジスタ領域0(p0)に「真」を、そうでない場合には「偽」を書き込む。ここで、プレディケイトレジスタ105に含まれる複数のプレディケイトレジスタ領域のうち、どのプレディケイトレジスタ領域に値を書き込むかは、演算命令1フィールドのプレディケイトレジスタ選択フィールド(Pフィールド)の一部を用いて指定されている。また、どのプレディケイトレジスタ領域に値を書き込むかの指定方法は、プレディケイトレジスタ選択フィールド(Pフィールド)の一部を用いる以外にも、例えば、命令フィールド(Iフィールド)の一部を用いて指定しても良い。

【0028】第二の命令語は減算命令(sub)である。この命令では、r1から即値"1"を減じ、その結果をr1に書き戻す処理を行う。但し、この命令語には「(p0)」という修飾語が前置されているので、もしp0が「真」であればこの減算命令は実行されるが、もしp0が「偽」であればこの減算命令は実行されない。

【0029】第三の命令語は加算命令(add)である。この命令では、データレジスタ領域2(r2)とr1とを加算し、r2に書き戻す処理を行う。

【0030】続いて、本実施の形態の別の動作の例について、図7、図8及び図9を用いて説明する。図7はC言語による別のプログラム例を、図8は図7のプログラムをプレディケイトレジスタ型の演算処理装置に与える形式にコンパイルした命令語列の例を示す。図8において、501は演算器1に与える命令語列、502は演算器2に与える命令語列である。ここでは簡単のため、演算器1と演算器2とが同じ速度で各命令語を処理するものとして説明する。

【0031】まず、演算器2は第一の命令語(add)により、r3とr4の加算を行い、結果をr5に格納する。この処理と同時に演算器1は第一の命令語(add)により、r1とr2の加算を行い、結果をr7に格納する。

【0032】続くステップで、演算器1は第二の命令語(mul)により、r7とr6の乗算を行い、結果をr7に格納する。同時に、演算器2は第二の命令語(xfer)により、データレジスタの値の転送を行う。この例では、データレジスタ202に含まれるデータレジスタ領域5(r5)が保持している値を、データレジスタ102に含まれるデータレジスタ領域5(r5)に書き込む。

【0033】次のステップで、演算器1と演算器2はともに第三の命令語(cmp.gt)により、比較命令を実行し、比較の結果をプレディケイトレジスタ領域0(p

0) にそれぞれ格納する。

【0034】さらに次のステップでは、演算器 1 は、p0 の値が「真」である場合、第四の命令語 (add) により r7 と即値 1 との加算を行い、結果を r7 に格納する。また同時に、演算器 2 は、p0 の値が「真」である場合、第四の命令語 (add) により r5 と即値 2 との加算を行い、結果を r5 に格納する。

【0035】次に、図 9 は、図 7 のプログラムを、本実施の形態のプレディケイトレジスタ型の演算処理装置に与える形式にコンパイルした命令語列の例を示す。図 9 において、511 は演算器 1 に与える命令語列、512 は演算器 2 に与える命令語列である。

【0036】この例の特徴は、命令語列 512 の第二命令語 (cmp.gt) に付加されている B パラメータである。このパラメータは、命令フィールドの中のプレディケイトレジスタ報知フィールド (B フィールド) に対応しており、他の演算器に対してプレディケイトレジスタの更新を報知 (Broadcast) するかどうかを決定する。この第二命令語を実行すると、演算器 2 のプレディケイトレジスタ領域 0 (p0) が保持している値が更新 20 されるだけでなく、他の演算器、この例では演算器 1、のプレディケイトレジスタ領域 0 (p0) の保持する値も更新される。この動作は、演算器 1 及び演算器 2 が持つセクタ 106 及び 206 (図 5) によって実現される。

【0037】通常動作の場合、プレディケイトレジスタ報知フィールド (B フィールド) の値は「偽」であり、この場合、セクタは、同じ演算器内の ALU が生成したプレディケイトレジスタへの書き込みデータを選択し、プレディケイトレジスタへ格納する。しかし、命令 30 語に B パラメータが指定された事によってプレディケイトレジスタ報知フィールド (B フィールド) の値が「真」であった場合、セクタは B フィールドが「真」である演算器から出力される書き込みデータを選択し、プレディケイトレジスタに格納する。

【0038】このような他の演算器に対して報知する事が可能な特徴を持つプレディケイトレジスタ更新手段を持つことによって、データレジスタの不要な転送処理を削減することが可能になる。

【0039】続いて、本実施の形態の別の動作の例について、図 10、図 11 及び図 12 に示す。図 10 は C 言語によるプログラム例を、図 11 は図 10 のプログラムをプレディケイトレジスタ型の演算処理装置に与える形式にコンパイルした命令語列の例を、図 12 は図 10 のプログラムを報知可能なプレディケイトレジスタ型の演算処理装置に与える形式にコンパイルした命令語列の例を、それぞれ示す。

【0040】この例では、演算処理装置は 3 つ以上の演算器を持つことを前提としている。図から判るように、上で述べた例と同様、報知機能を有効に利用する命令語 50

列を生成することで、より簡単な命令語列とすることが可能である。

【0041】次に、図 1 は、本発明の別の一実施の形態の演算処理装置の概略構成を示している。図 1 の実施の形態は、図 5 の実施の形態に対して、一つの演算器が内蔵している ALU の個数が異なる点 (図 5 では 1 個、図 1 では 2 個) と、プレディケイトレジスタへの書き込みデータを報知する際にセクタではなくバス 100 を介する点が、構成上の主な相違点である。以下の説明では、第一の実施の形態に対して構成または動作が異なる点について説明する。

【0042】本実施の形態では、命令語は演算命令 1 フィールド、演算命令 2 フィールド、演算命令 3 フィールド、演算命令 4 フィールド、の 4 つから構成される。各演算命令フィールドの構成要素は、第一の実施の形態と同じである。これらの演算命令フィールドのうち、演算命令 1 フィールドと演算命令 2 フィールドとは、第 1 の演算命令フィールドグループとして演算器 1 へ、演算命令 3 フィールドと演算命令 4 フィールドは、第 2 の演算命令フィールドグループとして演算器 2 へ、それぞれ供給される。各演算器は ALU をそれぞれ 2 つ内蔵しているため、命令レジスタ 101 及び 201 は、各々、同時にそれぞれ 2 つの命令フィールドを格納するように構成されている。

【0043】ALU からそれぞれ出力される、プレディケイトレジスタへ書き込む値は、それぞれの演算器内部のプレディケイトレジスタへ書き込まれる他に、プレディケイトレジスタ報知用のバスに出力することができる構成になっている。このバスを経由して、各 ALU によって出力されるプレディケイトレジスタへ書き込む値を、全演算器の指定されたプレディケイトレジスタ領域に同時に報知し、格納することができる。

【0044】このような構成を採ることによって、演算器の個数や、ALU の個数を増やした場合にも、多入力セクタによる信号遅延や、多くの信号線を独立に接続する為の実装面積の増大を、最小限に抑えたいうで実現することができる。

【0045】次に図 13 は、図 1 で示したプレディケイトレジスタ更新の手段について、詳しく示したものである。図 13 において、11 はどの ALU からの結果出力を報知するかを決定するプレディケイトレジスタ報知調停回路、12 は 11 で決定した結果出力を選択するプレディケイトレジスタ報知セクタ、13 は報知にかかわる例外発生回路、14 は例外発生回路から発生される例外信号、15 は例外信号 14 を受ける例外処理回路、16 は例外処理回路 15 から出力される演算停止信号、107 及び 207 はプレディケイトレジスタ 105 及び 205 に格納されている値を元にして ALU 103 及び 203 を動作させるかどうかを決める命令マスク手段 (論理積回路)、108 及び 208 はプレディケイトレジス

タ書き込み値が報知されている時にそれをプレディケイトレジスタに供給するプレディケイトレジスタ報知データ供給手段（論理和回路）、である。なお、図中の点線は、この説明には関係がないが、ALUとデータレジスタとを接続する回路を示している。

【0046】プレディケイトレジスタ報知調停回路11は、命令レジスタ101及び201に含まれるBフィールドの値を元に、どのALUの演算結果を報知するかを決定し、プレディケイトレジスタ報知セクタ12に伝える。決定のための方法は任意の方法で構わないが、例えば、演算器に予め設定した固定の優先順位によって決定する方法や、複数の演算器からの同時報知は資源衝突のエラーとして例外を発生する方法などが考えられる。例外発生位回路13は、そのような例外信号14を発生するための回路であり、この例外信号14に応じて例外処理回路14は演算停止信号16を各演算器に供給する。これにより、各演算器の演算が停止され、リセット待ち状態となる。

【0047】プレディケイトレジスタ報知セクタ12は、調停回路11の決定に基づき、必要なら各演算器からの演算結果を報知する。

【0048】報知されたデータは、プレディケイトレジスタ報知データ供給手段108及び208によって、プレディケイトレジスタ105及び205に供給される。プレディケイトレジスタ報知データ供給手段108及び208は、単純な論理和を求める回路によって構成される。この回路は、必ずしも論理和を求める手段である必要はなく、例えば、報知データが供給されているときには報知データを、供給されていないときには演算器内部のALUからのデータを、選択的に供給する選択回路でも良い。

【0049】プレディケイトレジスタ105及び205は、複数の記憶領域を持つことを想定している。プレディケイトレジスタへの値の格納の際に、どのプレディケイトレジスタ領域に対して値を格納するかは、命令レジスタ101及び201に格納されているオペランドの値によって、決定される。また、プレディケイトレジスタからの値の読み出しの際に、どのプレディケイトレジスタ領域から値を読み出すかも、命令レジスタ101及び201に格納されているオペランドの値によって、決定される。これら、プレディケイトレジスタ領域の選択に関わる情報は、命令レジスタのPフィールド、もしくはIフィールドに格納される。

【0050】以上の実施の形態では、演算器相互間でのプレディケイトレジスタの値の選択は、プレディケイトレジスタへの書き込みデータを選択する方法によって実現していた。すなわち、どのALUの出力をプレディケイトレジスタに書き込むかを切り換えることにより、プレディケイトレジスタの値の選択を行っていた。しかし、プレディケイトレジスタの値の選択は、プレディケ

イトレジスタからの読み出しデータを選択する方法を用いて行うこともできる。

【0051】この場合の動作の例について、図15に示す。図15は、図10のプログラムを、読み出し値を報知可能なプレディケイトレジスタ型の演算処理装置に与える形式にコンパイルした命令語列の例を示す。ここでは、参照すべきプレディケイトレジスタを示すために、プレディケイトレジスタを示す命令語に、演算器の番号を指定する構成としている。すなわち、例えば、演算器2の第三の命令語の先頭部の「(p0:1)」は、演算器1のプレディケイトレジスタ領域1(p1)を参照すべきことを規定している。

【0052】図14は、このような機能を実現する本発明のさらに別の実施の形態の演算処理装置の概略構成を示している。すなわち、この実施の形態は、図1の実施の形態に対して、プレディケイトレジスタの書き込み値を報知するのではなく、プレディケイトレジスタからの読み出し値を報知する点が、構成上の主な相違点である。以下の説明では、主として、前出の実施例に対して構成または動作が異なる点について説明する。

【0053】各ALUから出力される、プレディケイトレジスタへ書き込む値は、それぞれの演算器内部のプレディケイトレジスタへ書き込まれる。各プレディケイトレジスタから読み出される値は、プレディケイト報知用のバス140に出力される。このバスを經由して、各ALUはどの演算器が格納しているプレディケイトレジスタの値も参照することができる。どのプレディケイトレジスタの値を参照するかは、図15で前述したように、プレディケイトレジスタを参照する命令語の中に指定された演算器の番号により決定される。

【0054】このような構成を取ることによって、演算器の個数や、ALUの個数を増やした場合にも、多入力セクタによる信号遅延や、多くの信号線を独立に接続するための実装面積の増大を、最小限に抑えた上で実現することができる。

【0055】プレディケイトレジスタの報知の方法を、図1のような構成とするか、図14のような構成とするかは、ソフトウェアの構造に影響を与える点以外、任意である。

【0056】また、本発明は上述した実施の形態のみに限定されるものではなく、要旨を逸脱しない範囲で適宜内容を変更して実施することができる。

【0057】例えば、実施の形態中では、演算器は最大2個、1演算器当たりのALUは最大2個、として記述しているが、この値はなんら制約となるものではなく、必要に応じて演算器やALUの数を増減することが可能である。

【0058】また実施の形態中では、命令語の構成や命令フィールドの構成について示しているが、この構成自体もここに示した例に限らず、自由な構成で実施可能で

10

20

30

40

50

ある。

【0059】また実施の形態中では、プレディケイトレジスタは、特定のバスでのみ読み書きが行われるように記述しているが、これは説明の簡単のためであり、例えば、任意のデータレジスタと任意のプレディケイトレジスタとの間で値を転送するといった構成となっても差し支えない。

【0060】

【発明の効果】以上述べたように本発明によれば、1命令語中に複数の演算命令フィールドを持ち、複数の演算の処理を並列に実行する演算処理装置において、複数の演算を実行するための複数の演算器の相互間での、演算結果の転送処理の回数を削減することで、処理プログラムのステップ数を小さくすることが可能になる。

【0061】また、演算器相互間で演算結果データを転送するのに必要な信号線数は、データレジスタのビット幅に対してプレディケイトレジスタのビット幅が小さいために、より狭い面積での回路実装が可能であり、演算器相互間の転送をより高速に行うことが可能である。

【0062】以上の利点により、演算処理装置全体の実性能を向上させることができるという効果がある。

【図面の簡単な説明】

【図1】 本発明の他の一実施の形態の演算処理装置の概略構成を示すブロック図である。

【図2】 C言語によるプログラム例を示す説明図である。

【図3】 従来型の演算処理装置に与える形式にコンパイルした例を示す説明図である。

【図4】 プレディケイトレジスタ型の演算処理装置に与える形式にコンパイルした例を示す説明図である。

【図5】 本発明の一実施の形態の演算処理装置の概略構成を示すブロック図である。

【図6】 本発明の一実施の形態での命令語を示す説明図である。

【図7】 C言語によるプログラム例を示す説明図である。

【図8】 プレディケイトレジスタ型の演算処理装置に与える形式にコンパイルした命令語列の例を示す説明図である。

【図9】 実施の形態のプレディケイトレジスタ型の演算処理装置に与える形式にコンパイルした命令語列の例

を示す説明図である。

【図10】 C言語によるプログラム例を示す説明図である。

【図11】 プレディケイトレジスタ型の演算処理装置に与える形式にコンパイルした命令語列の例を示す説明図である。

【図12】 報知可能なプレディケイトレジスタ型の演算処理装置に与える形式にコンパイルした命令語列の例を示す説明図である。

【図13】 プレディケイトレジスタ更新手段の概略構成を示すブロック図である。

【図14】 本発明のさらに他の実施の形態の演算処理装置の概略構成を示すブロック図である。

【図15】 読み込み値を報知可能なプレディケイトレジスタ型の演算処理装置に与える形式にコンパイルした命令語列の例を示す説明図である。

【符号の説明】

1…演算器1、2…演算器2、3…命令キャッシュ、4…データキャッシュ、11…プレディケイトレジスタ報知調停回路、12…プレディケイトレジスタ報知セクタ、13…例外発生回路、14…例外信号、15…例外処理回路、16…演算停止信号、100…バス、101…命令レジスタ、102…データレジスタ、103…ALU、104…ALU、105…プレディケイトレジスタ、106…セクタ、107…命令マスク手段、108…プレディケイトレジスタ報知データ供給手段、140…バス、201…命令レジスタ、202…データレジスタ、203…ALU、204…ALU、205…プレディケイトレジスタ、206…セクタ、207…命令マスク手段、208…プレディケイトレジスタ報知データ供給手段、501…演算器1向け命令列、502…演算器2向け命令列、511…演算器1向け命令列、512…演算器2向け命令列、601…命令語、611…演算命令1フィールド、612…演算命令2フィールド、621…プレディケイトレジスタ選択フィールド(Pフィールド)、622…プレディケイトレジスタ報知フィールド(Bフィールド)、623…命令フィールド(Iフィールド)、631…プレディケイトレジスタ選択フィールド(Pフィールド)、632…プレディケイトレジスタ報知フィールド(Bフィールド)、633…命令フィールド(Iフィールド)。

【図2】

図2

```

001  if( a > 2 )
002  {
003      a = a - 1;
004  }
005  b = b + a;

```

【図3】

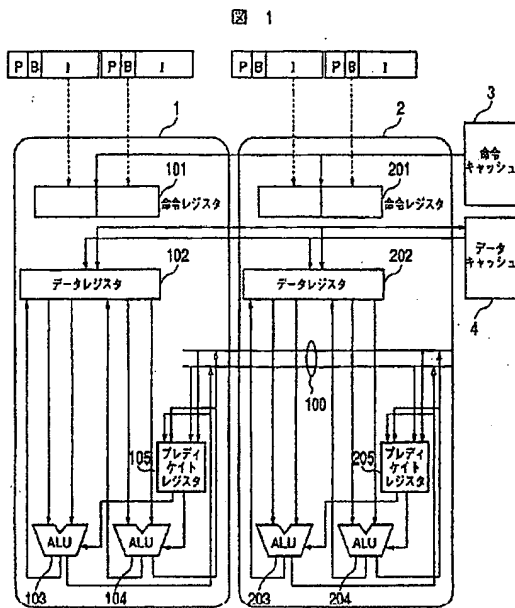
図3

```

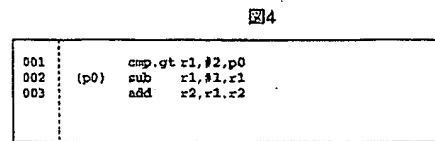
001      cmp    r1,#2
002      ble    $1
003      sub    r1,#1,r1
004  $1:      add    r2,r1,r2

```

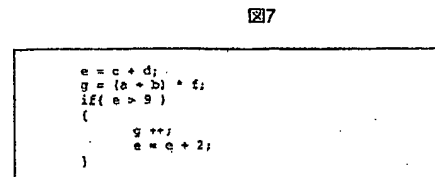

【図 1】



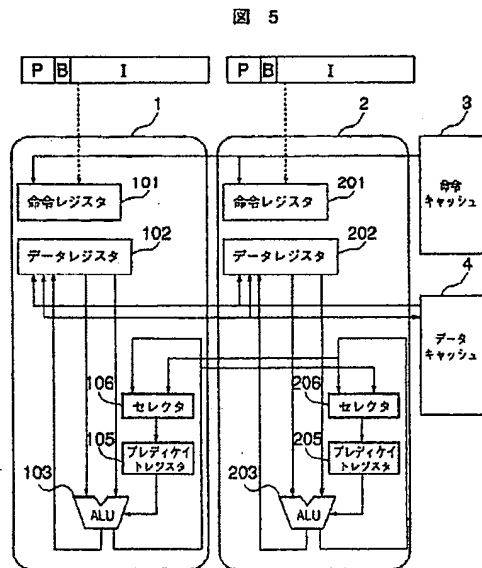
【図 4】



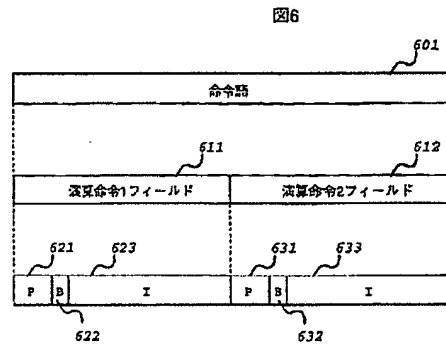
【図 7】



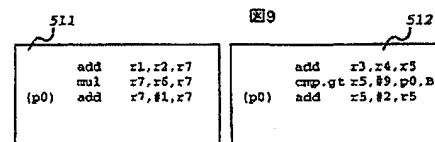
【図 5】



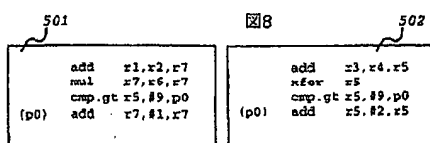
【図 6】



【図 9】

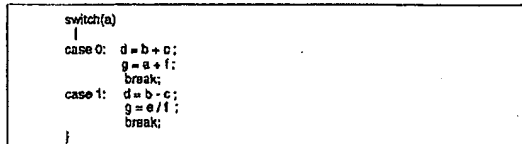


【図 8】



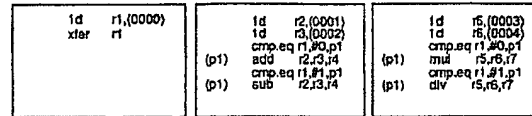
【図10】

図 10



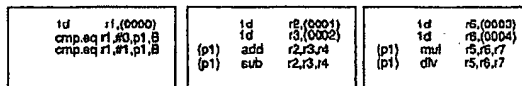
【図11】

図 11



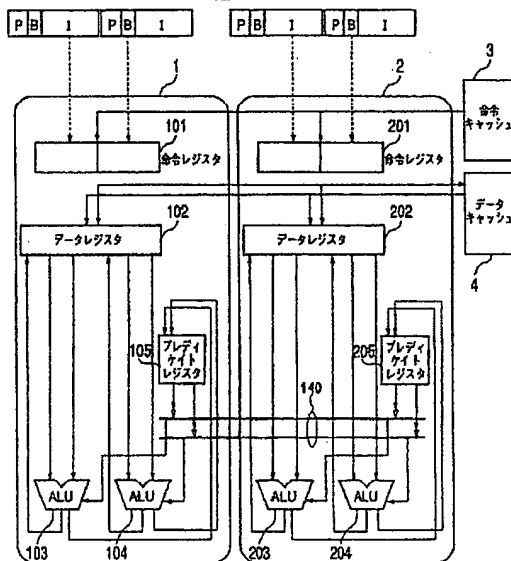
【図12】

図 12



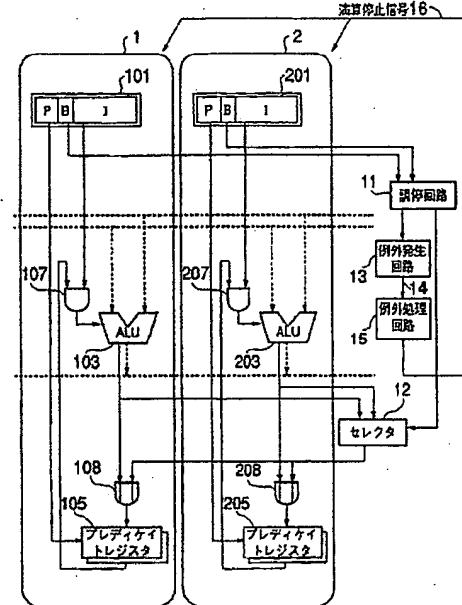
【図14】

図 14



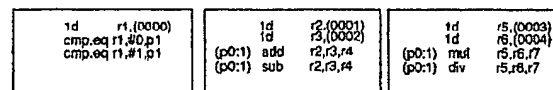
【図13】

図 13



【図15】

図 15



フロントページの続き

(72)発明者 野尻 徹

神奈川県川崎市麻生区王禅寺1099番地 株
式会社日立製作所システム開発研究所内

(72)発明者 西岡 清和

神奈川県川崎市麻生区王禅寺1099番地 株
式会社日立製作所システム開発研究所内

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☒ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.